



# Contribution d'une approche syntaxique dans la segmentation d'image

Jean Camillerapp, Ivan Leplumey

## ► To cite this version:

Jean Camillerapp, Ivan Leplumey. Contribution d'une approche syntaxique dans la segmentation d'image. [Rapport de recherche] RR-0331, INRIA. 1984. inria-00076226

**HAL Id: inria-00076226**

**<https://inria.hal.science/inria-00076226>**

Submitted on 24 May 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



CENTRE DE RENNES

IRISA

Rapports de Recherche

N° 331

**CONTRIBUTION  
D'UNE APPROCHE SYNTAXIQUE  
DANS  
LA SEGMENTATION D'IMAGE**

**Jean CAMILLERAPP  
Ivan LEPLUMEY**

**Septembre 1984**

Institut National  
de Recherche  
en Informatique  
et en Automatique

Domaine de Voluceau  
Rocquencourt  
B.P. 105

78153 Le Chesnay Cedex  
France

Tél (3) 954 90 20

Campus Universitaire de Beaulieu  
Avenue du Général Leclerc  
35042 - RENNES CÉDEX  
FRANCE  
Tél. : (99) 36.20.00  
Télex : UNIRISA 95 0473 F

## CONTRIBUTION D'UNE APPROCHE SYNTAXIQUE DANS LA SEGMENTATION D'IMAGE

Jean CAMILLERAPP - Ivan LEPLUMEY

Publication Interne n°231

Juillet 1984

46 pages

Résumé : Dans un bon nombre d'images, la valeur de la luminosité en un point n'est pas un critère suffisant pour assurer la segmentation. Dans le cas cependant où l'on peut caractériser les objets par leur contraste avec leur environnement, nous proposons une définition syntaxique de ces objets.

Cette définition permet de construire les automates et donc les algorithmes qui assurent la reconnaissance. L'intérêt d'une telle approche réside dans la faible puissance de calcul nécessaire, et dans l'utilisation des informations contenues dans un contexte dont la taille n'est pas définie a priori, pour prendre les décisions de segmentation.

Abstract : Generally, the luminosity intensity of a pixel is not a sufficient criteria for image segmentation. The purpose of this report is to show that, however, when objects can be characterized with their environment, it is possible to achieve image segmentation. The method relies on a syntactic definition of the objects and enables to construct automata and algorithms for their recognition.

The interest of this approach is two-fold : (i) the segmentation decision is derived from a large and variable context, as opposed, for instance, to more conventional methods which only take into account a local and fix-sized context ; and (ii) the algorithms are very cheap and consume little processing power.

## I - INTRODUCTION

Les méthodes syntaxiques sont employées couramment dans les problèmes d'analyse de scènes et de reconnaissance de formes ([Cam1], [Cam2], [Cam3], [Cuz], [Fu1], [Fu2], [Gon], [Kac1], [Kac2]).

On s'aperçoit cependant que si à un certain niveau de reconnaissance, ces méthodes semblent avoir fait leur preuve, elles ne sont pratiquement pas utilisées au niveau le plus bas, celui de la segmentation de l'image à partir de la luminosité des pixels, niveau qui correspond à la détection des contours.

Ce domaine est presque exclusivement celui des opérateurs numériques de type gradient (Sobel, Roberts...) ou laplacien ([Bout], [Gas], [Mait1], [Pel], [Ram], [Set], [Tim] ) par exemple.

Ceux-ci sont relativement coûteux, mais surtout ils n'opèrent qu'une transformation des valeurs numériques de l'image sans aucune extraction de structure. Celle-ci doit être faite dans une étape ultérieure.

Les travaux allant dans le sens du remplacement des opérateurs numériques par des méthodes syntaxiques en détection de contours existent cependant ([Cam2], [Cam3], [Cuz], [Mait2]). Amin KACED ([Kac1], [Kac2]) définit les k-formes qui sont un recodage des variations locales de luminosité. L'intérêt d'une approche qui s'appuie plus sur le formalisme des grammaires, et notamment des grammaires d'attributs est de fournir à l'opérateur de segmentation une information qui provient d'un contexte beaucoup plus important et surtout dont la taille n'est pas déterminée a priori, ce qui est le cas avec les opérateurs numériques ou avec les k-formes. HOROWITZ (|Hor|, |Mic|) propose de segmenter le signal (électrocardiogramme) en zone de monotonie croissante ou décroissante, et d'extraire des paramètres de classification.

Dans un premier temps, nous avons voulu apprécier les résultats qu'il était possible d'obtenir avec ce type d'approche. Les comparaisons de vitesse et d'efficacité avec d'autres méthodes non pas encore été effectuées. Nous donnerons cependant une évaluation du nombre d'opération élémentaires par pixel, qui permet de se faire une idée des performances possibles.



## II - CARACTERISTIQUES DES IMAGES ANALYSEES

Les recherches en traitement d'images ont vite montré qu'il était illusoire de vouloir un détecteur de contour universel, mais qu'il fallait s'adapter à la nature de l'image et des objets qui la composent ; le rapport signal/bruit, la vitesse de variation de l'orientation des contours, le pas d'échantillonnage... sont des paramètres importants.

Nous allons donc commencer par définir les caractéristiques approximatives des images étudiées tout en sachant que nous préciserons ces caractéristiques par la suite.

Pour simplifier l'exposé, nous supposons que la valeur de la luminosité d'un point d'un objet est localement supérieure à la luminosité d'un point du fond, le cas inverse s'en déduisant immédiatement.

### Définition approximative des objets

Ces objets peuvent avoir des natures sensiblement différentes les uns des autres :

- objets de grande taille et fortement contrastés
- objets fins (6 à 8 pixels de large) et très faiblement contrastés (3 à 5 niveaux de digitalisation)
- objets proches les uns des autres ou de forme complexe, et qui induisent des perturbations locales de luminosité.

### III - CADRE EXPERIMENTAL

L'étude s'appuie sur des radiographies du réseau sanguin cérébral. En attendant les images fournies par un appareil de radiologie numérique, nous avons travaillé sur des images numérisées par un béliographe. Ces images ont une bonne définition ( $970 \times 966$ ), mais elles sont très bruitées.

Tous ces résultats présentés portent sur une sous-image  $350 \times 350$ , mais les algorithmes sont prévus pour pouvoir traiter des images de grande taille, c'est-à-dire impossible à mettre en totalité en mémoire centrale.

Des variantes de ces méthodes ont été utilisées sur d'autres images d'origine très différente.

#### IV - LE SEUILLAGE ABSOLU

Définition : Soit  $V$  une valeur numérique, soit  $I$  une image, nous appellerons seuillage absolu, l'opération qui consiste à valider tous les points de l'image dont la luminosité  $I(x, y)$  est supérieure ou égale à  $V$  appelé seuil absolu (fig. 1).

L'application sur l'image d'étude d'un seuil absolu  $V$ , montre que quelle que soit cette valeur  $V$ , il est impossible de séparer correctement les objets du fond, c'est-à-dire de trouver une valeur  $V$  telle que tout point d'un objet ait sa luminosité supérieure ou égale à  $V$ , et que tout point du fond ait sa luminosité inférieure à  $V$  (fig. 1).

Dans le formalisme de la théorie de la décision à deux classes, cela revient à dire que la règle de décision " $I(x, y)$  affecté à objet si  $I(x, y) > V$ " a une probabilité d'erreur trop élevée, ou encore que le paramètre de luminosité n'est pas suffisamment informant. La probabilité  $P$  (objet/luminosité) est très voisine de 0.5 sur un intervalle de valeurs de la luminosité qui comporte beaucoup d'occurrences.

L'histogramme de la fenêtre d'étude montre la répartition des luminosités (fig. 2). Les objets se trouvent approximativement dans l'intervalle [199-249] de l'échelle des luminosités. Le fond lui occupe la place [191-208]. Si l'intervalle de recouvrement n'est pas ici très important [199-208], il faut remarquer que c'est dans cet intervalle qu'il y a le plus grand nombre des valeurs, et que c'est également dans cet intervalle que la validité de l'affectation à le plus d'importance. De plus, deux objets proches peuvent, avec un seuil absolu, être considérés comme un seul objet (fig. 3, fig. 4).

Compte tenu de ces remarques, il faut utiliser un seuil local que nous préciserons par la suite, il correspondra à la règle de décision " $I(x, y)$  affecté à objet si  $I(x, y) > V(x, y)$ ".

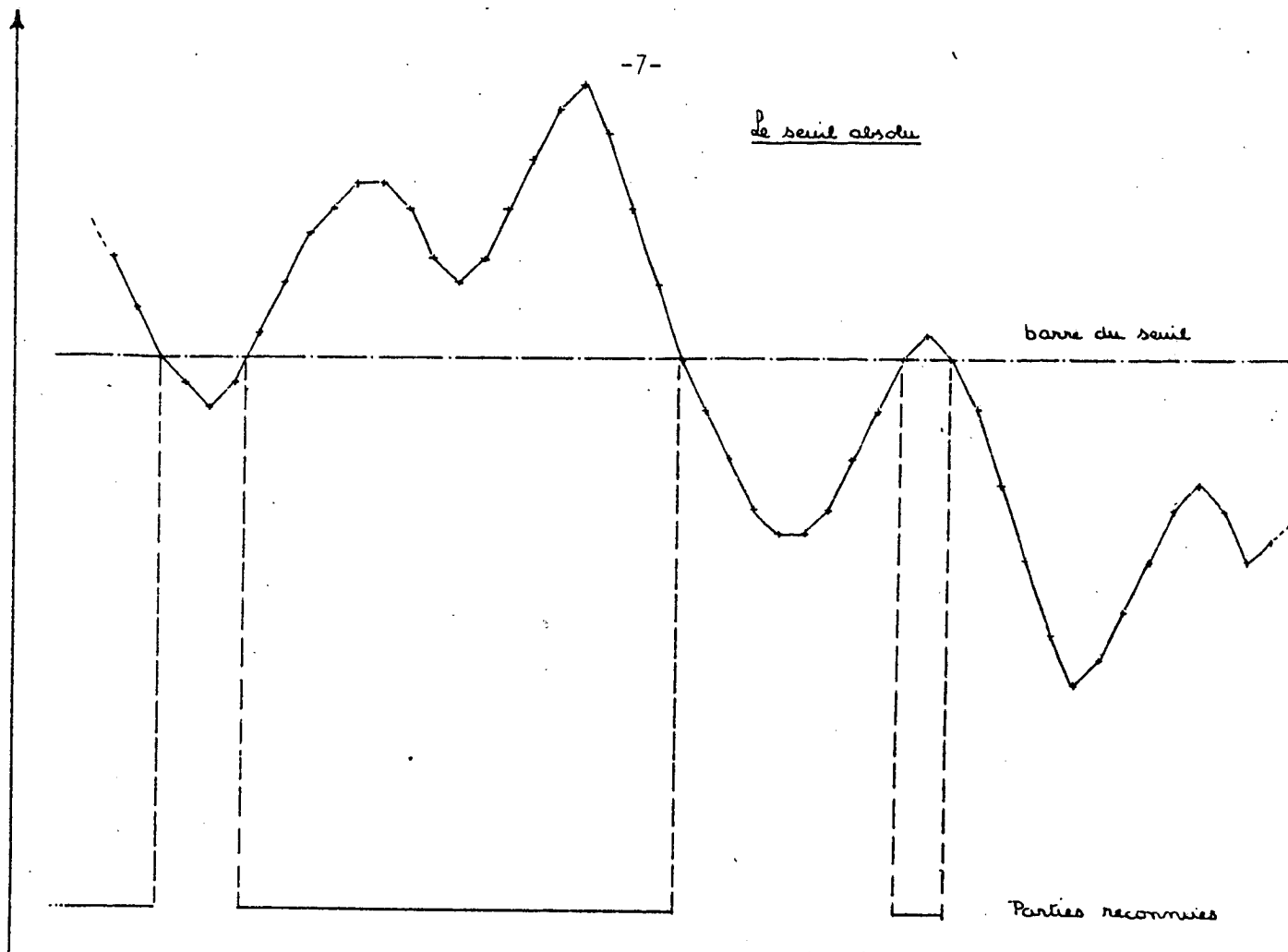


Figure 1 - Le seuil absolu

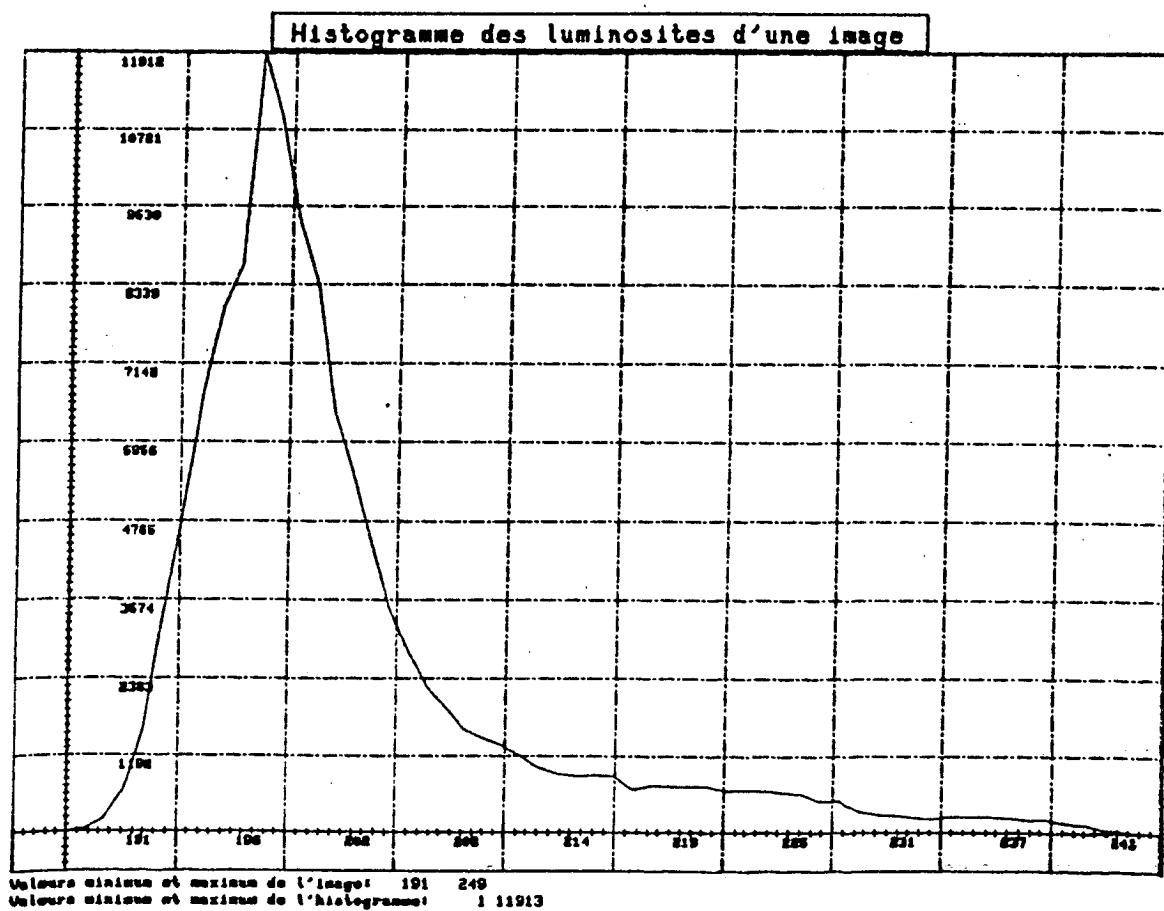


Figure 2 - Histogramme de la fenêtre d'analyse.



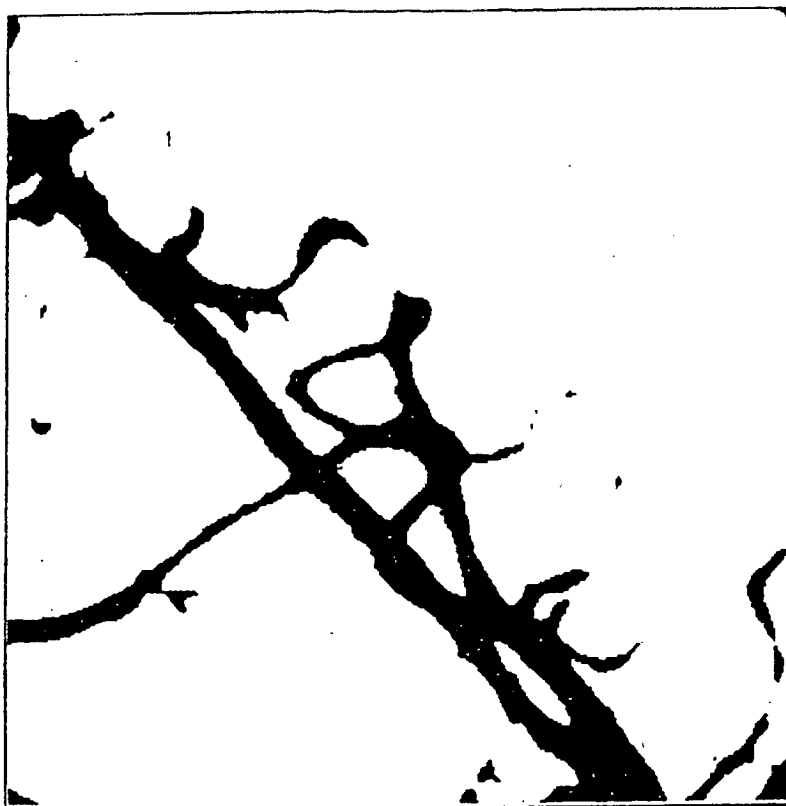


Figure 3 - Seuil absolu.

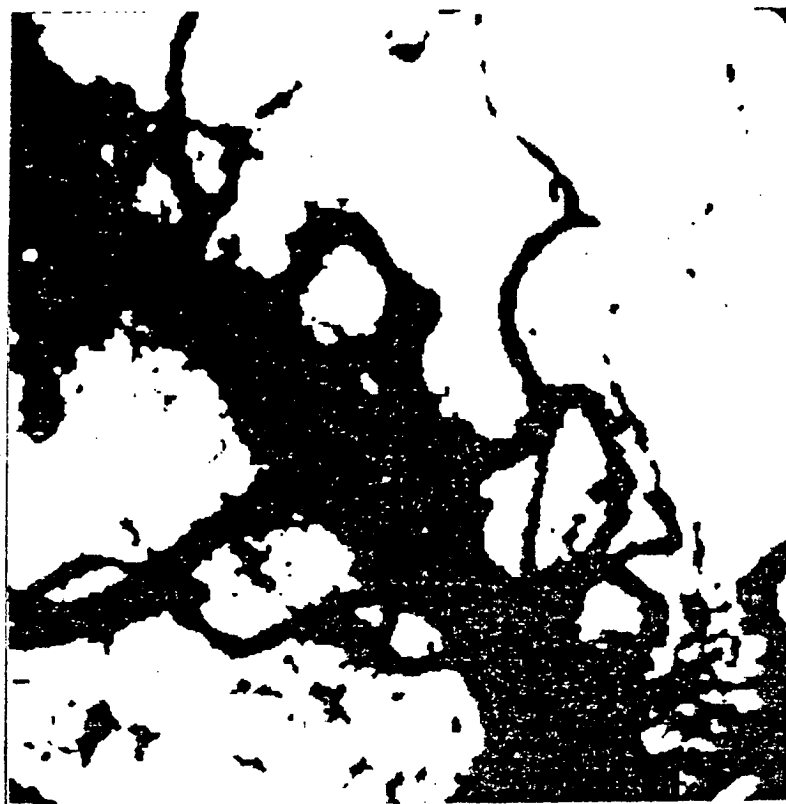


Figure 4 - Seuil absolu.

## V - RECONNAISSANCE MONODIMENSIONNELLE

Le premier niveau de reconnaissance se fait en segmentant une ligne de l'image en intervalles selon certaines caractéristiques de son profil de luminosité.

### V.1. Définition d'une forme caractéristique

Définition : Une forme caractéristique est composée de 2 parties, une partie gauche consistant en une montée de luminosité, une partie droite consistant en une descente de luminosité (Sch.1).

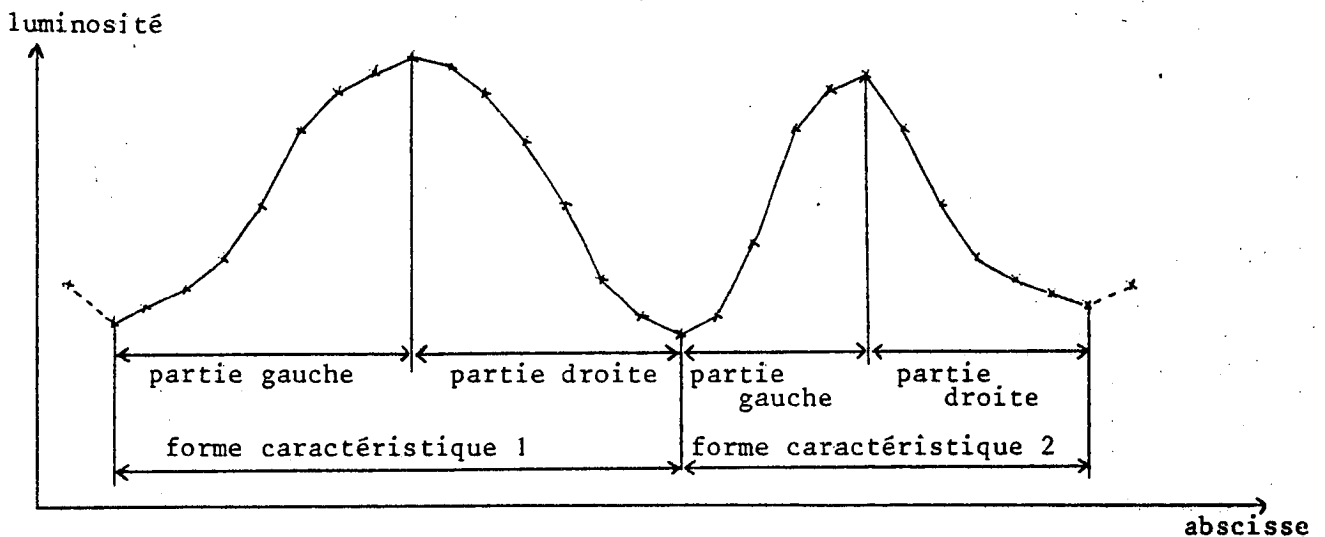


Schéma 1 - Profil selon une ligne

Nous associerons à chaque intervalle, sept paramètres numériques :

- numéro de ligne où se trouve la forme caractéristique
- numéro de colonne de début de forme caractéristique (début de la montée) et luminosité associée
- numéro de colonne de fin de forme caractéristique (fin de la descente) et luminosité associée
- numéro de colonne du maximum de luminosité de la forme caractéristique et valeur de ce maximum.

La segmentation des lignes en formes caractéristiques et l'ex-

traction des paramètres associés peuvent être modélisées par une grammaire d'attributs modèle à la fois syntaxique et sémantique. Ceci résout le problème de la définition algorithmique des objets pour lequel l'oeil a une définition "physiologique" qu'il était jusqu'à présent difficile de transcrire.

## V.2. Rappel sur les grammaires

Nous rappelons ici quelques notions fondamentales, nous renvoyons à [Liv], [Pag], [Knu], [Mic] pour des explications plus détaillées.

### Hiérarchie de Chomsky

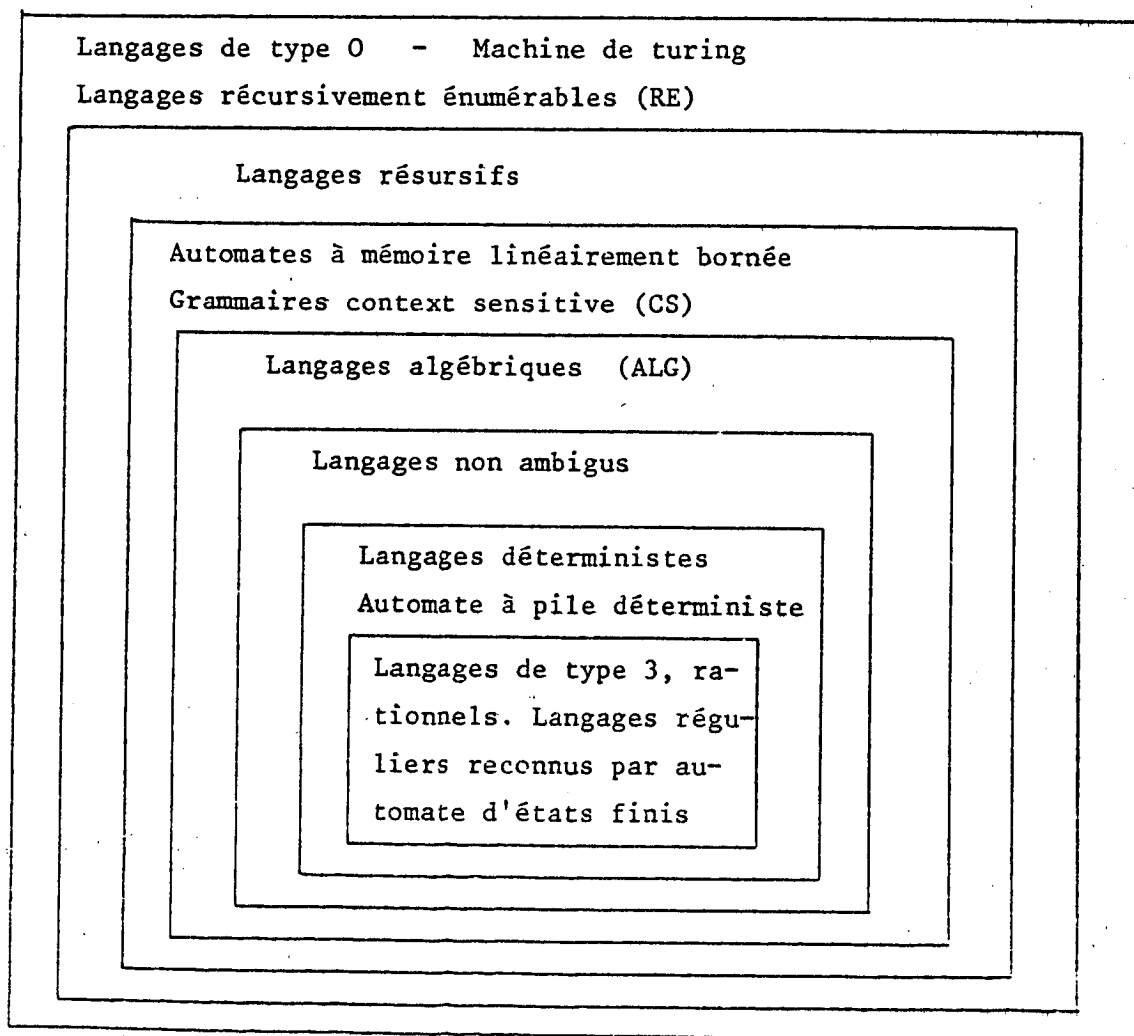


Schéma 2 - Hiérarchie de Chomsky

Chomsky a proposé de classer les langages en fonction de leur puissance et de leur difficulté de reconnaissance. On peut traduire cette classification dans le schéma 2.

Définition : Une grammaire algébrique  $G$  est la donnée de 4 éléments :

- 1 - un ensemble fini  $\Sigma$ , dit alphabet terminal
- 2 - un ensemble fini  $V$ , dit alphabet auxiliaire ou non-terminal
- 3 - un symbole distingué  $S \in V$ , appelé axiome
- 4 - une partie finie  $P \subseteq V \times (\Sigma \cup V)^*$  dite ensemble des règles de production.

Définition : Pour qu'une grammaire soit régulière, il faut que ses règles de production soient du type

$$A \rightarrow yB$$

$$\text{ou } A \rightarrow x$$

$$\text{avec } \{x, y\} \in \Sigma \quad \{A, B\} \in V$$

Le langage associé à ce type de grammaire est appelé langage régulier. Ce type de langage a l'avantage d'être reconnu (ou engendré) par des automates d'états finis, qui demandent une quantité finie de mémoire ([Sim]).

Définition : Une grammaire algébrique est propre si et seulement si  $P \subseteq V \times ((\Sigma \cup V)^* / (V \cup \{1\}))$ .

Définition formelle de la méthode des attributs ([Liv])

Considérons un langage  $L$  défini par une grammaire algébrique  $G = (\Sigma, V, ::=, S)$ . Les seules conditions imposées à cette grammaire sont que cette dernière soit réduite supérieurement et inférieurement et que l'axiome  $S$  apparaisse uniquement en partie gauche des règles de  $G$ . Rappelons qu'une grammaire est dite réduite inférieurement et supérieurement si tout non-terminal peut se dériver en un mot du langage engendré et si tout non-terminal apparaît dans un mot dérivant de l'axiome. Pour définir une sémantique de  $L$  par la méthode des attributs, on utilise les outils suivants :

- Pour chaque symbole,  $Y$  de  $V$ , on se donne un ensemble éventuellement vide de symbole  $At(Y)$  réunion de deux sous-ensembles disjoints  $Ats(Y)$  et  $Ath(Y)$ . les éléments de  $Ats(Y)$  sont appelés attributs synthétisés de  $Y$  et ceux de  $Ath(Y)$  attributs hérités de  $Y$ . On impose de plus que  $Ath(S) = \emptyset$  et

par convention si  $a$  est un symbole terminal de  $G$ , on pose  $At(a) = \emptyset$ .

- A chaque attribut  $a$  est associé un ensemble  $V_a$  appelé ensemble des valeurs de l'attribut  $a$ . Pour chaque  $Y$  de  $V \cup \Sigma$ , on note  $V_Y$  l'ensemble

$$\prod_{a \in At(Y)} V_a.$$

- Pour chaque règle  $r$  de  $G$  de la forme  $Y_0 ::= Y_1 Y_2 \dots Y_n$ , on se donne des applications qui permettent de calculer les attributs synthétisés de  $Y_1, Y_2, \dots, Y_n$  et de calculer les attributs hérités de chaque  $Y_i$ , en fonction des attributs de  $Y_j$ . Pour chaque règle  $r$ , on dispose donc des applications suivantes :

$$F_0[r] : V_{Y_1} \times \dots \times V_{Y_n} \longrightarrow \prod_{a \in Ats(Y_0)} V_a$$

$$1 \leq i \leq n \quad F_i[r] : V_{Y_0} \times V_{Y_1} \times \dots \times V_{Y_n} \longrightarrow \prod_{a \in Ath(Y_i)} V_a.$$

### V.3. Description de la grammaire de segmentation d'une ligne en formes caractéristiques

#### Propriétés :

La grammaire développée pour l'extraction des primitives est une grammaire algébrique propre, c'est une grammaire régulière, donc reconnaissable par un automate d'état fini.

#### L'alphabet terminal

L'alphabet terminal est composé de 3 éléments  $\{m, p, d\}$  désignant respectivement montée, plat et descente résultant d'un transcodage à partir de deux points consécutifs d'une même ligne.

Soient deux points consécutifs, d'indice colonne  $j$  et  $j+1$ , à ces deux points, on associe un élément de l'alphabet terminal suivant la table de transcodage (Table 1).

montée	luminosité $(j) <$ luminosité $(j+1)$	m
plat	luminosité $(j) =$ luminosité $(j+1)$	p
descente	luminosité $(j) >$ luminosité $(j+1)$	d

Table 1

L'utilisation d'un élément de  $\Sigma$  correspond à celle d'un quadruplet  $(j, \text{luminosité}(j), j+1, \text{luminosité}(j+1))$ . Ce sont ces quatre valeurs

qui servent au calcul de la plupart des attributs de la grammaire.

### L'alphabet auxiliaire V

L'alphabet auxiliaire V est composé de quatre éléments {ligne, I, M, D} qui ont comme valeur sémantique celle reportée dans la table ci-dessous.

Nom	Signification
ligne	Représente une ligne d'image
I	Représente l'état d'amorçage. Il correspond à la volonté d'ignorer d'éventuels objets qui seraient coupés par le bord gauche de l'image. Cela se traduit par l'élimination des caractères p et d en début de ligne
M	Représente les montées de luminosité
D	Représente les descentes de luminosité

Table 2

### L'axiome

L'axiome correspond à l'élément {ligne} de l'alphabet auxiliaire.

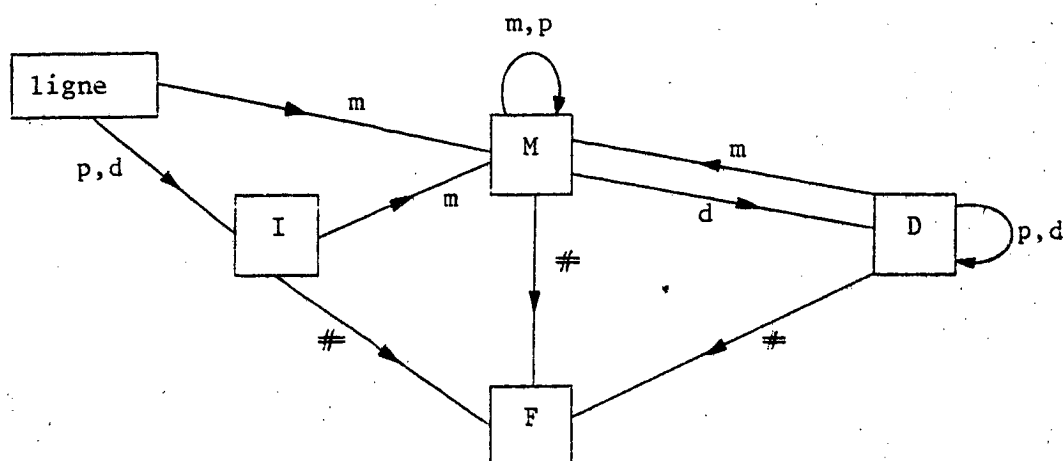
### Règles de production

ligne  $\rightarrow$  pI | dI | mM  
 I  $\rightarrow$  pI | dI | mM | #  
 M  $\rightarrow$  pM | mM | dD | #  
 D  $\rightarrow$  pD | dD | mM | #

Nous avons décrit précédemment l'alphabet terminal  $\Sigma$ , nous ajoutons un élément particulier # qui correspond au caractère de fin de ligne. Le quadruplet (j, luminosité (j),  $\emptyset$ ,  $\emptyset$ ) est associé à ce caractère.

De plus, la règle {ligne  $\rightarrow$  #} a été omise, étant donné que l'on suppose qu'une ligne d'image possède plus de 2 points.

On peut également représenter ces règles de production par le graphe des états ci-contre :



Etat initial : ligne

Etat final : F

Graphe des états

Attributs employés

Les tables 3 et 4 décrivent d'une part les associations attributs-élément alphabet auxiliaire, d'autre part la signification des attributs utilisés.

Elément de l'alphabet auxiliaire	Attributs associés
ligne	- -
I	{Nlig}
M	{Nlig, objet}
D	{Nlig, objet, correct}

Table 3

Dans la suite, nous appellerons "Archivage d'objet", l'archivage de "objet" en vue d'un traitement ultérieur. La table 5, donne la correspondance des règles syntaxiques et sémantiques.

Nom de l'attribut	Signification
Nlig	Attribut hérité Numéro de la ligne en cours d'analyse

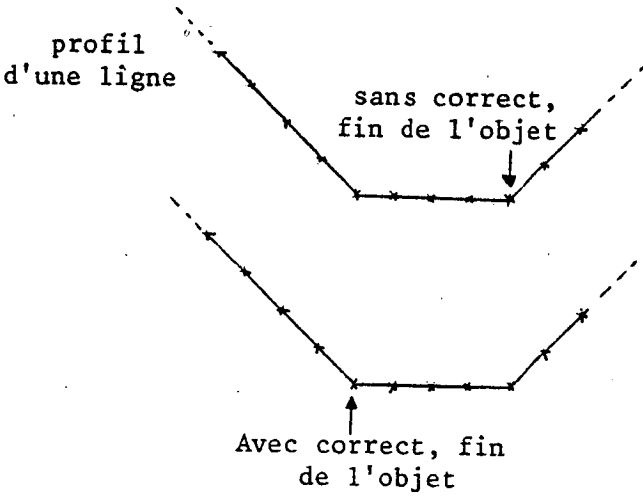
Objet	<p>Attribut hérité</p> <p>Structure composée de 7 éléments, décrivant la forme caractéristique en cours d'analyse</p> <ul style="list-style-type: none"> <li>- Objet.Lig : numéro de la ligne sur laquelle se trouve la forme caractéristique</li> <li>- Objet.L1 : luminosité du début de la forme</li> <li>- Objet.P1 : position du début de la forme</li> <li>- Objet.L2 : luminosité maximum trouvée dans la forme</li> <li>- Objet.P2 : position du maximum de luminosité de la forme</li> <li>- Objet.L3 : luminosité de la fin de la forme</li> <li>- Objet.P3 : position de la fin de la forme</li> </ul>
Correct	<p>Attribut hérité</p> <p>Valeur de correction de objet.P3 en cas de plat en fin d'objet</p> 

Table 4

Règles syntaxiques	Règles sémantiques
ligne → pI   dI	Nlig(I) = ind. de ligne



ligne $\rightarrow$ mM	$Nlig(M) = \text{ind. de ligne}$ $Objet(M).P1 = j$ $Objet(M).L1 = \text{luminosité (j)}$ $Objet(M).Lig = Nlig(M)$
$I \rightarrow pI \mid dI$	$Nlig(I_2) = Nlig(I_1)$
$I \rightarrow \#$	-
$I \rightarrow mM$	$Nlig(M) = Nlig(I)$ $Objet(M).P1 = j$ $Objet(M).L1 = \text{luminosité (j)}$ $Objet(M).Lig = Nlig(M)$
$M \rightarrow pM \mid mM$	$Nlig(M_2) = Nlig(M_1)$ $Objet(M_2) = Objet(M_1)$
$M \rightarrow \#$	-
$M \rightarrow dD$	$Correct(D) = 0$ $Nlig(D) = Nlig(M)$ $Objet(D) = Objet(M)$ $Objet(D).P2 = j$ $Objet(D).L2 = \text{luminosité (j)}$
$D \rightarrow dD$	$Nlig(D_2) = Nlig(D_1)$ $Objet(D_2) = Objet(D_1)$ $Correct(D_2) = 0$
$D \rightarrow pD$	$Nlig(D_2) = Nlig(D_1)$ $Objet(D_2) = Objet(D_1)$ $Correct(D_2) = correct(D_1) + 1$
$D \rightarrow \#$	$Objet(D).P3 = j - correct(D)$ $Objet(D).L3 = \text{luminosité (j)}$ Archivage d'objet(D)
$D \rightarrow mM$	$Nlig(M) = Nlig(D)$ $Objet(D).P3 = j - correct(D)$ $Objet(D).L3 = \text{luminosité (j)}$ Archivage d'objet(D) $Objet(M).Lig = Nlig(M)$ $Objet(M).P1 = j$ $Objet(M).L1 = \text{luminosité (j)}$

Table 5

Comme nous l'avons déjà dit, la traduction de ces règles en langage algorithmique est pratiquement immédiate. De plus, la simplicité de ces règles est telle que l'on peut parfaitement envisager leur implantation sur une petite unité centrale.

#### V.4. Position du maximum de luminosité d'une forme caractéristique

Soit la forme caractéristique (partie du profil d'une ligne satisfaisant aux conditions déjà précisées) définie sur le schéma 3.

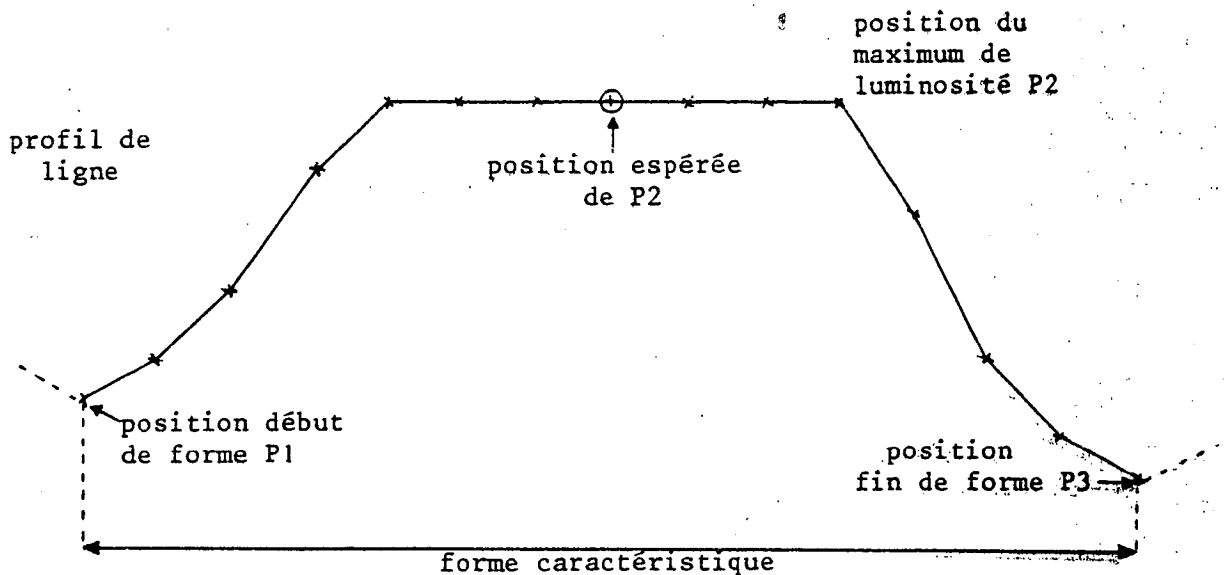


Schéma 3

La valeur affectée à l'attribut P2 en cas de plat au sommet d'une forme est discutable. On pourrait préférer placer P2 au milieu du plat du sommet.

Pour répondre à cette objection, il suffit d'apporter les modifications suivantes à la grammaire :

1 - Ajout à l'élément M de l'alphabet auxiliaire d'un attribut hérité correct, qui a comme signification une correction sur P2 en cas de plat au sommet d'un objet.

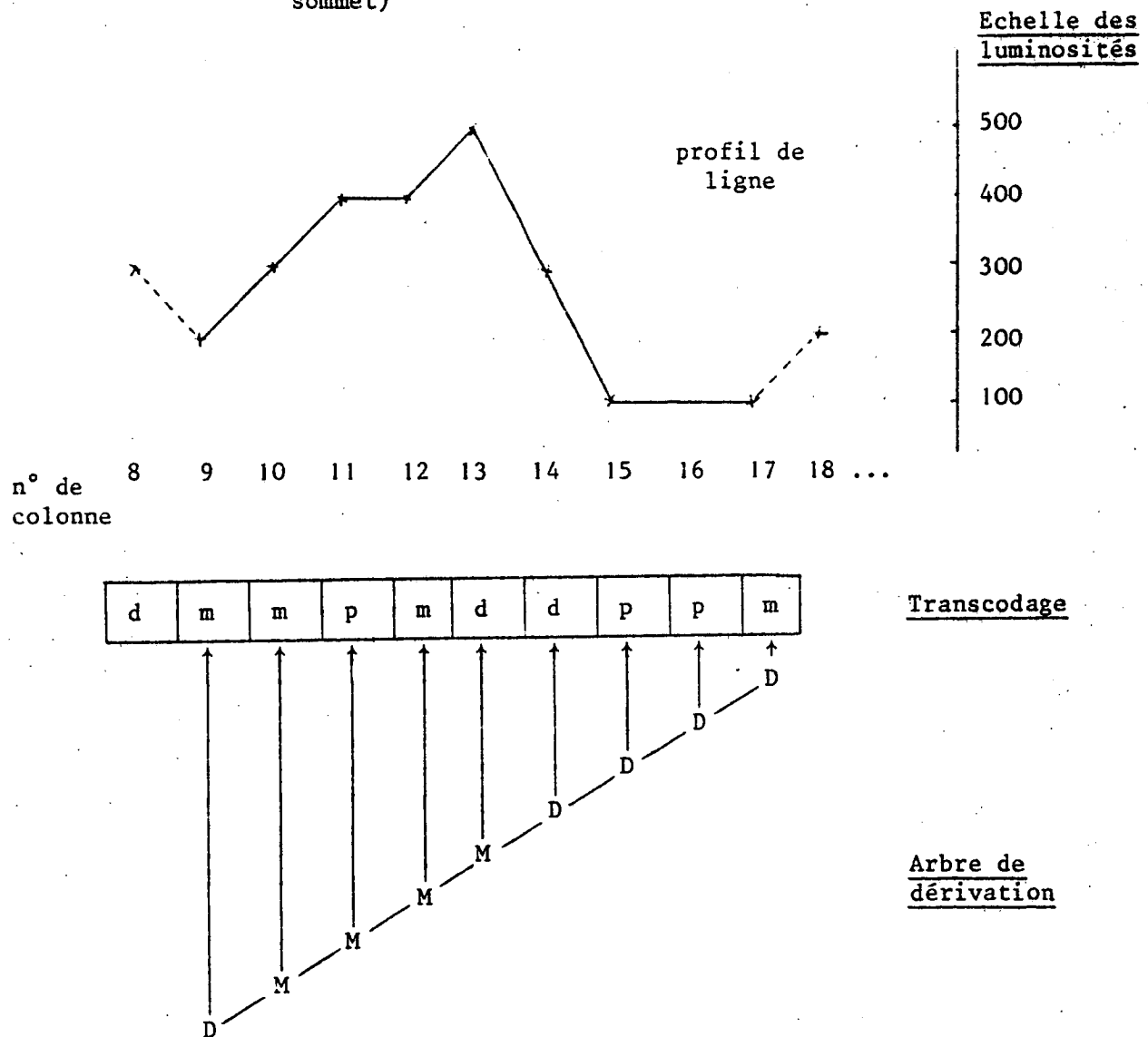
2 - Modifications des règles sémantiques associées à certaines règles syntaxiques (table 6). Cette table reprend les règles syntaxiques qui ont eu leurs règles sémantiques modifiées.

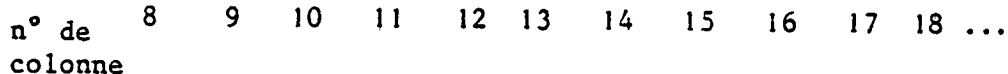
Règles syntaxiques	Règles sémantiques
$I \rightarrow mM$	$Nlig(M) = Nlig(I)$ $Objet(M).Lig = Nlig(M)$ $Objet(M).P1 = j$ $Objet(M).L1 = \text{luminosité } (j)$ <u><math>Correct(M) = 0</math></u>
$M \rightarrow pM$	$Nlig(M_2) = Nlig(M_1)$ $Objet(M_2) = Objet(M_1)$ <u><math>Correct(M_2) = Correct(M_1) + 1</math></u>
$M \rightarrow mM$	$Nlig(M_2) = Nlig(M_1)$ $Objet(M_2) = Objet(M_1)$ <u><math>Correct(M_2) = 0</math></u>
$M \rightarrow dD$	$correct(D) = 0$ $Nlig(D) = Nlig(M)$ $Objet(D) = Objet(M)$ <u><math>Objet(D).P2 = j - correct(M)/2</math></u> $Objet(D).L2 = \text{luminosité } (j)$
$D \rightarrow mM$	$Nlig(M) = Nlig(D)$ <u><math>Correct(M) = 0</math></u> $Objet(D).P3 = j - correct(D)$ $Objet(D).L3 = \text{luminosité } (j)$ Archivage d'objet(D) $Objet(M).Lig = Nlig(M)$ $Objet(M).P1 = j$ $Objet(M).L1 = \text{luminosité } (j)$
$\text{ligne} \rightarrow mM$	$Nlig(M) = \text{ind. de ligne}$ $Objet(M).Lig = Nlig(M)$ $Objet(M).P1 = j$ $Objet(M).L1 = \text{luminosité } (j)$ <u><math>Correct(M) = 0</math></u>

Table 6

Comme le phénomène "plat en sommet" est peu important sur les images analysées, dans toute la suite de l'étude nous n'apportons pas cette correction sur la position des maximum de luminosité.

V.5. Exemple de fonctionnement (sans correction sur plat en sommet)





d	m	m	p	m	d	d	p	p	m
---	---	---	---	---	---	---	---	---	---

## Transcodage

x	x	x	x	x	x	x	x	x	Nlig
				0	0	1	2	2	Correct

Valeur des attributs  
après exécution des  
règles sémantiques

(excepté pour la  
dernière colonne de  
Objet qui représente  
l'objet archivable)

[illegible]

## VI - SELECTION ET SEUIL DE CONTRASTE

Les grammaires précédentes aboutissent à supprimer les effets de bord (droit et gauche), et à segmenter le reste de la ligne en formes caractéristiques qui sont archivées au fur et à mesure.

Cependant, parmi les formes caractéristiques archivées, il y en a un certain nombre qui ne sont pas significatives, car pas assez contrastées. Nous allons donc leur appliquer un opérateur de sélection. La forme en cours d'analyse ne sera archivée pour un traitement ultérieur que si elle satisfait un critère.

### Définition du contraste gauche d'un objet (fig. 5)

Soit X, un objet, nous appellerons contraste gauche de l'objet X, la valeur :

$$CG(X) = (Objet(X).L2 - Objet(X).L1).$$

### Définition du contraste droit d'un objet (fig. 5)

Soit X, un objet, nous appellerons contraste droit de l'objet X, la valeur :

$$CD(X) = (Objet(X).L2 - Objet(X).L3).$$

### Définition du critère de sélection

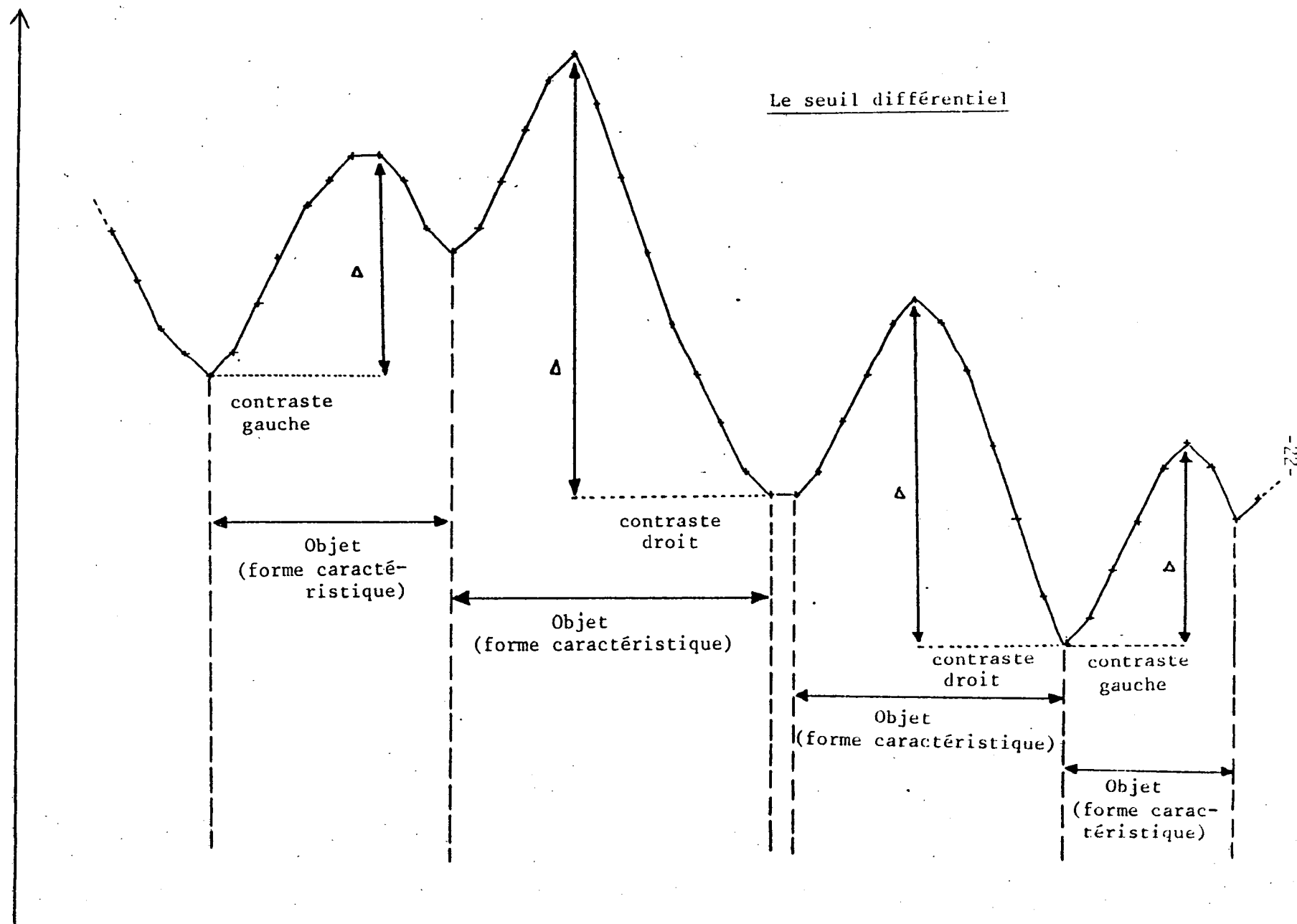
Un objet sera archivable s'il satisfait le critère suivant :  
 $(CG(X) \geq V) \vee (CD(X) \geq V).$

Pour une image donnée, le seuil V est constant:

Si dans son esprit, l'opérateur d'extraction est voisin d'un opérateur de dérivation, on note que la décision est prise sur une monotonie qui constitue un contexte beaucoup plus grand que les 3 à 5 pixels classiques des opérateurs de dérivation. Le fait que la décision s'appuie sur une fraction de ligne dont la taille n'est pas définie a priori constitue une des forces du procédé.

Les seules modifications à effectuer sur la grammaire initiale consistent à transformer les archivages d'objets en archivages conditionnels. Nous donnons dans le tableau 7 les modifications à apporter.

Figure 5 - Le seuil différentiel.



Règles syntaxiques	Règles sémantiques
$D \rightarrow \#$	<p>Objet(D).P3 = j - correct (D)  Objet(D).L3 = luminosité (j)  <u>Si</u> CG(D) &gt;= V <u>ou</u>  CD(D) &gt;= V  <u>alors</u> archivage d'objet (D)  <u>fsi</u></p>
$D \rightarrow mM$	<p>Nlig(M) = Nlig(D)  Objet(D).P3 = j - correct (D)  Objet(D).L3 = luminosité (j)  <u>Si</u> CG(D) &gt;= V <u>ou</u>  CD(D) &gt;= V  <u>alors</u> Archivage d'objet(D)  <u>fsi</u>  Objet(M).Lig = Nlig(M)  Objet(M).P1 = j  Objet(M).L1 = luminosité (j)</p>

Table 7

L'ordre d'exécution des règles sémantiques pour une règle syntaxique donnée conditionne le bon fonctionnement de la grammaire.

Le passage de cette grammaire sur une image avec un seuil de contraste de 5 donne déjà des résultats intéressants (fig. 6).





Figure 6 - Seuil différentiel : 5

## VII - AMELIORATION DE LA DETECTION

Les résultats fournis par la grammaire précédente sur l'image initiale avec suppression des objets dont les caractéristiques sont inférieures à un certain seuil possèdent un certain nombre de défauts.

- 1 - Le résultat reste très bruité.
- 2 - A cause de ce bruit, un contraste plus faible est difficilement acceptable.
- 3 - Flottement du contour des vaisseaux.
- 4 - Les vaisseaux sensiblement horizontaux, c'est-à-dire parallèles au sens de balayage de l'image, ne sont pas détectés.

### VII.1. Résolution des problèmes de bruit

Les résultats dépendent des extréma et de leur contexte, et pour résoudre ce problème de bruit nous n'avons pas voulu poursuivre dans le domaine des règles de réécriture ([Cam2], [Cuz], [Quig]) (Schéma 4). En effet, nous aurons par la suite à analyser des images obtenues à partir d'autres capteurs qui ne présentent peut-être pas ces caractéristiques, et comme il est possible de réduire les extrêmes parasites par un filtrage coupant bien les hautes fréquences (bruit sur 2 ou 3 échantillons), nous avons considéré le filtrage comme une opération auxiliaire nécessaire et non comme une partie essentielle de l'analyse.

Plusieurs filtres passe-bas ont été essayés ; le choix de celui-ci n'est pas critique, les résultats après l'opérateur syntaxique sont assez robustes par rapport aux coefficients du filtre.

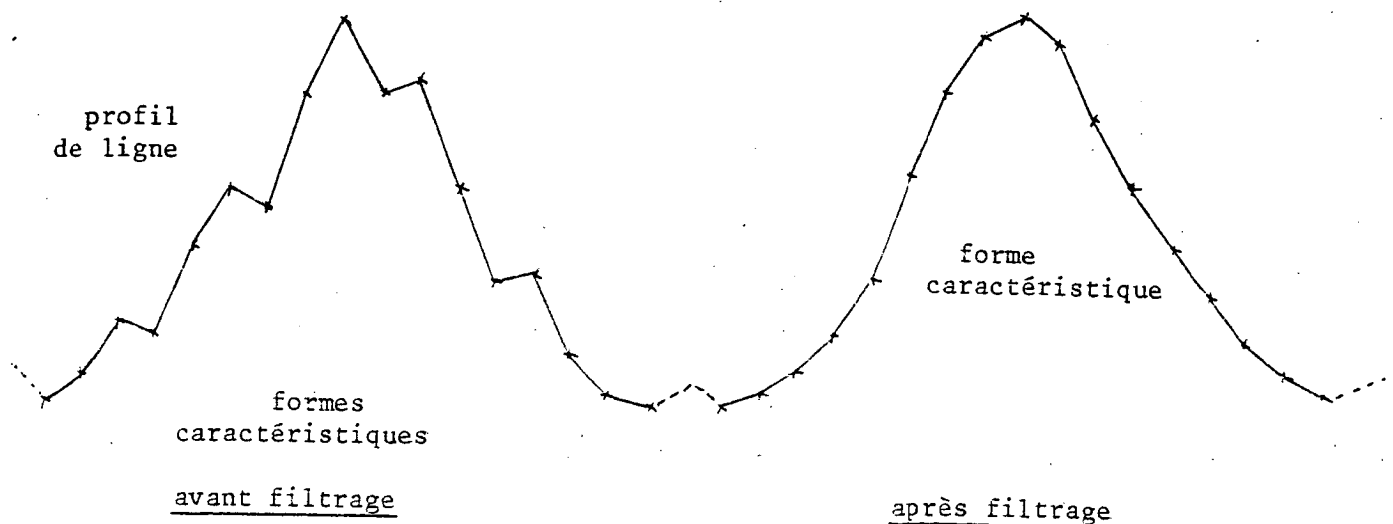


Schéma 4

Nous avons utilisé un filtre bidimensionnel 5 x 5 défini par les coefficients :

1	2	2	2	1
2	4	4	4	2
2	4	4	4	2
2	4	4	4	2
1	2	2	2	1

Ce filtre a l'avantage d'être décomposable, c'est-à-dire qu'il correspond à un filtrage ligne suivi d'un filtrage colonne par le filtre monodimensionnel de coefficients 1 2 2 2 1 (fig. 7)

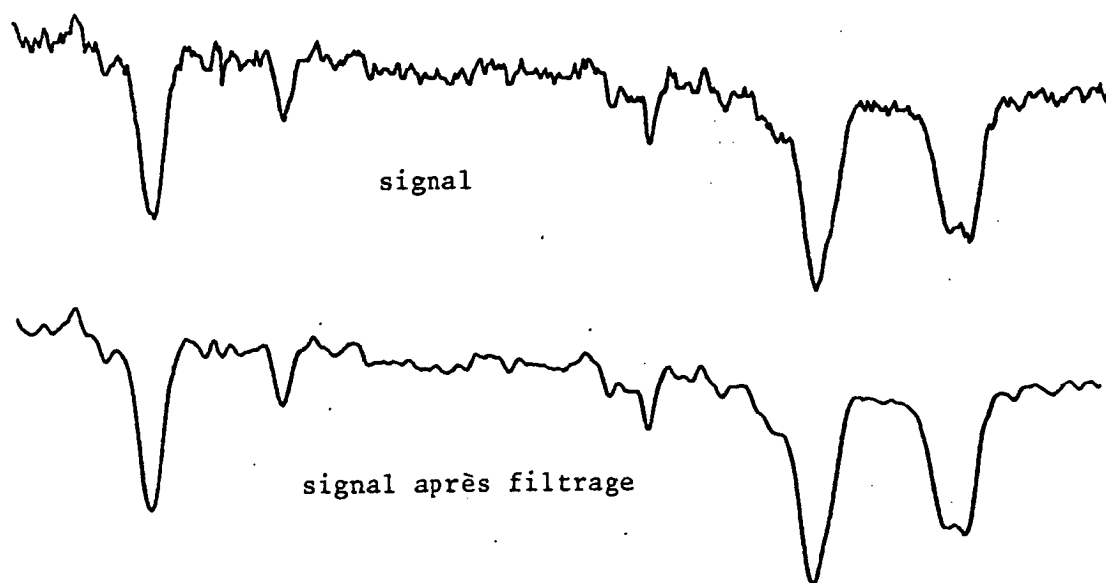


Figure 7 - Effets du filtre 1 2 2 2 1.

Cette remarque permet de réduire considérablement le coût du filtrage. Le filtrage a été fait en laissant tous les coefficients du filtre entier, ce qui permet de travailler dans l'ensemble N, l'échelle des luminosités est donc multipliée par 64, multiplication qui devra aussi être appliquée au seuil de contraste. Ainsi un seuil de 320 dans l'image filtrée correspond à un seuil de 5 dans l'image initiale. Le pendant de cette dernière remarque est qu'un pixel est codé sur un nombre supérieur de bits et donc que l'image codée prend plus de place. Dans une chaîne opérationnelle, cette contrepartie a peu d'importance, car il est possible de combiner filtrage et analyse, l'augmentation de place ne porte alors que sur les lignes conservées en mémoire centrale pour leur traitement.

La figure 8 présente les résultats avec un seuil de 320 (5 x 64), elle est donc comparable à la figure 6. On peut maintenant sans problème abaisser ce seuil (fig. 9, avec seuil 256, soit 4 dans l'échelle de départ).



Figure 8 - Après filtrage, seuil différentiel : 320



Figure 9 - Après filtrage, seuil différentiel : 256

## VII.2. Détection des objets horizontaux

Le détecteur de formes caractéristiques effectue un balayage ligne de l'image. C'est un détecteur anisotrope : tout vaisseau parallèle à la direction de balayage ne sera pratiquement pas détecté. Plutôt que de considérer comme [QUINQ] le traitement des objets horizontaux comme un traitement d'exception, nous avons essayé la chaîne de traitement suivante :

Tous les vaisseaux verticaux sont détectés, mais pas les vaisseaux horizontaux

Extraction des formes caractéristiques de l'image initiale

Transposition de l'image initiale

Tous les vaisseaux horizontaux sont détectés, mais pas les vaisseaux verticaux.

Extraction des formes caractéristiques de l'image initiale transposée

Tous les vaisseaux sont détectés (cf.fig. 10)

Représentation graphique des 2 extractions, superposées dans un même référentiel

Cette chaîne de traitement, facile à mettre en oeuvre, donne au détecteur la propriété d'isotropie, mais à un coût de traitement élevé à cause d'une transposition et d'un double passage du détecteur ligne, sans compter le coût d'entrées/sorties, l'image n'étant pas supposée tenir en mémoire centrale. Cette chaîne a permis cependant d'évaluer rapidement la qualité du détecteur isotrope.

Les résultats globaux sont représentés en figure 10.

La figure 11 est un agrandissement du coin supérieur droit où se trouvent à la fois des objets horizontaux et verticaux. Elle permet d'apprécier la complémentarité de la phase verticale et de la phase horizontale du détecteur. Une remarque cependant, en de multiples endroits il y a une redondance de l'information, ainsi un vaisseau à 45° sera détecté par la phase verticale, mais aussi par la phase horizontale du détecteur.

Les résultats obtenus justifient la recherche d'une implémentation des grammaires qui fassent les 2 extractions lors d'un balayage ligne.



Figure 10 - Visualisation des formes caractéristiques horizontales et verticales au seuil différentiel 320.

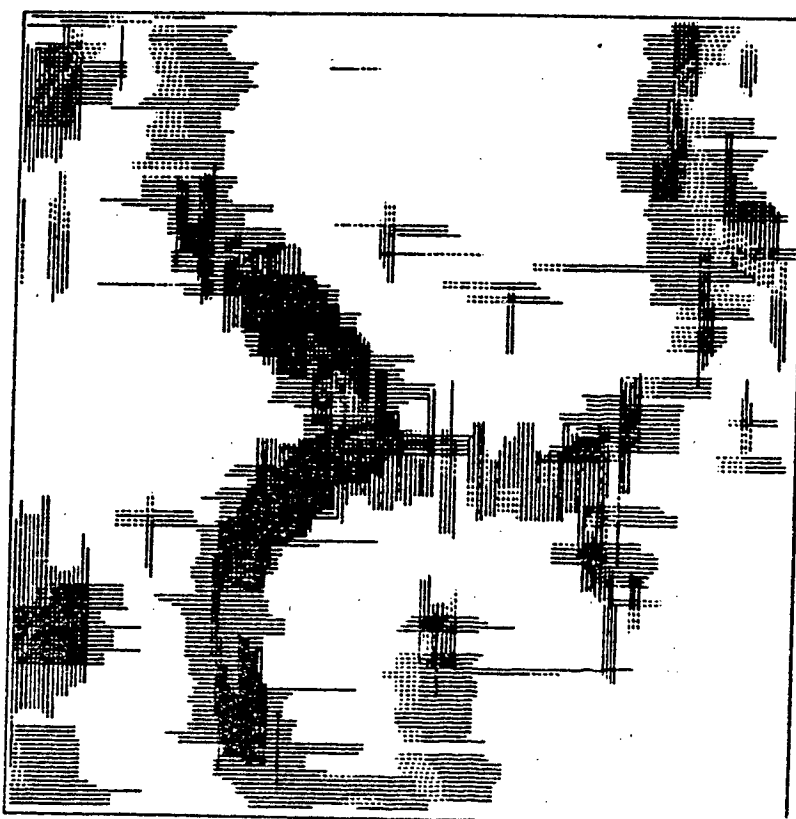
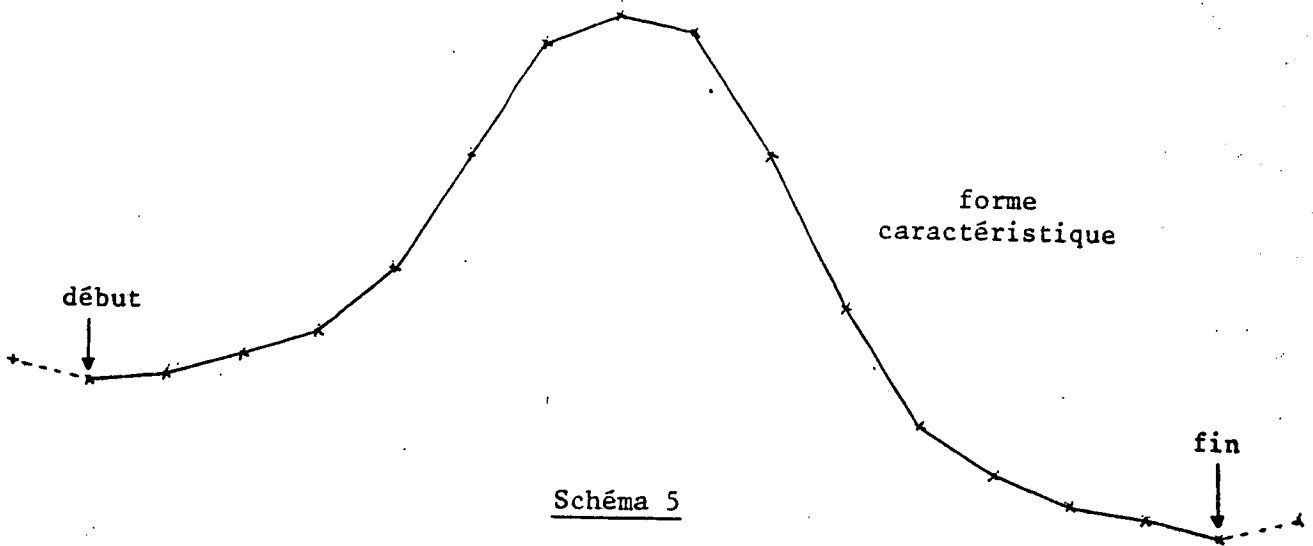


Figure 11 - Agrandissement du coin supérieur droit de la fig. 10.

### VII.3. Lissage du contour des vaisseaux

Le phénomène de flottement est dû au fait que le début et la fin d'une forme caractéristique sont très sensibles au bruit car ils se trouvent dans une région où la luminosité varie peu (schéma 5). Il faut donc définir d'une façon plus robuste le début et la fin d'un objet :



L'idée est d'appliquer aux formes caractéristiques une coupe à mi-hauteur sur chaque flanc (Fig. 12).

#### Définition du début d'une forme

Soit P1 le début d'une forme définie par les grammaires précédentes et L1 la luminosité associée, soit P2 la position du maximum de luminosité de cette même forme et L2 ce maximum, P'1 début de l'objet par coupe à mi-hauteur sera défini comme suit :

$$\begin{aligned} \text{luminosité}(P'1-1) &< (\text{luminosité}(P2) + \text{luminosité}(P1))/2 \\ \text{et } \text{luminosité}(P'1) &\geq (\text{luminosité}(P2) + \text{luminosité}(P1))/2 \end{aligned}$$

Ce qui est équivalent à :

$$\begin{aligned} \text{luminosité}(P'1-1) &< (L2 + L1)/2 \\ \text{et } \text{Luminosité}(P'1) &\geq (L2 + L1)/2 \end{aligned}$$

#### Définition de la fin d'une forme

De même si P3 est la fin d'une forme définie au sens des anciennes grammaires et L3 la luminosité associée, P2 et L2 ayant les mêmes significations que précédemment, P'3 fin de l'objet par coupe à mi-hauteur

Les formes caractéristiques  
représentées avec coupe à  
mi-hauteur

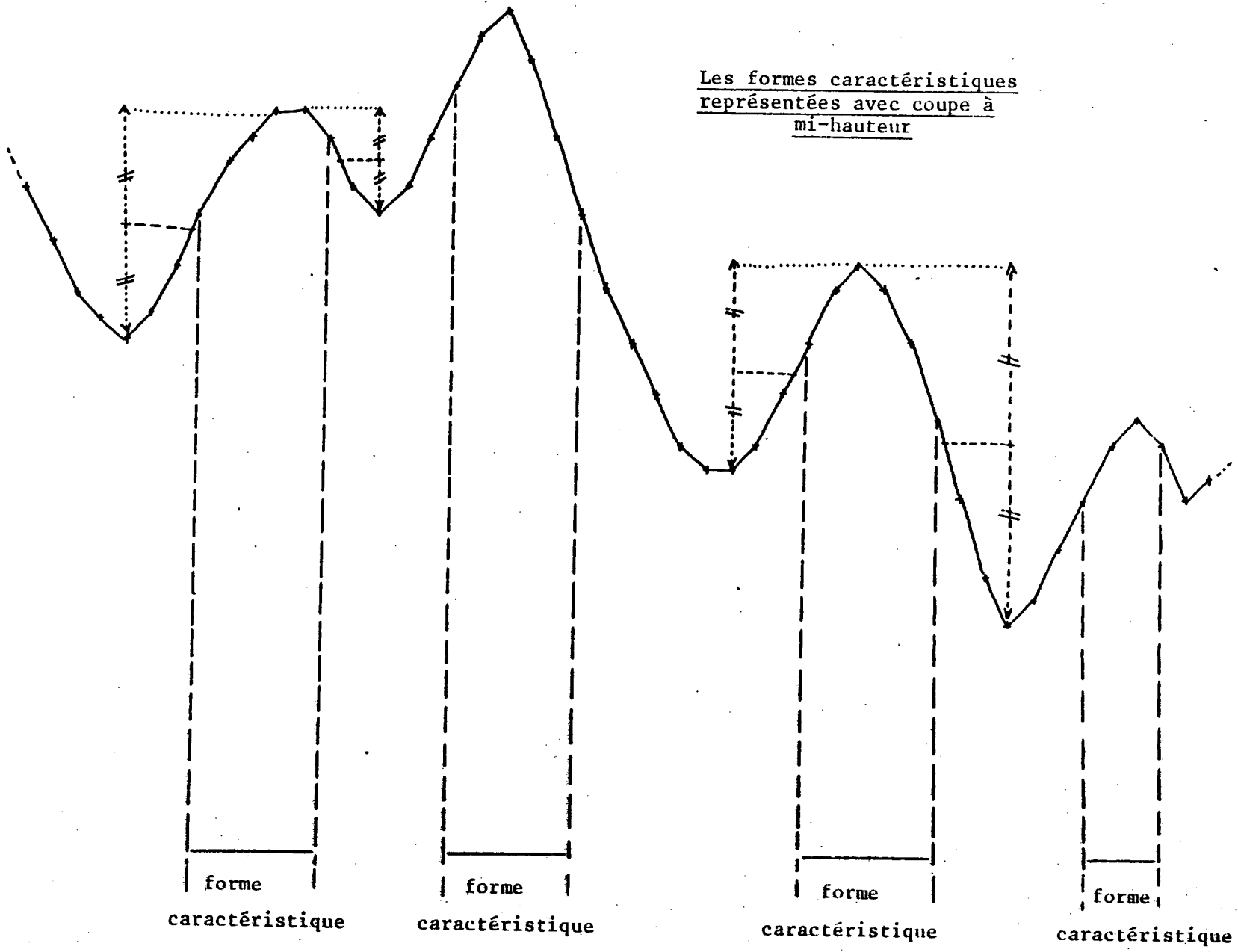


Figure 12 - La coupe à mi-hauteur.



sera défini comme suit :

$$\text{luminosité}(P'_{3+1}) < (L_2 + L_3)/2$$

et luminosité(P'3)  $\geq (L2 + L3)/2$

La grammaire de détection et sélection de formes caractéristiques est transformée pour qu'elle accepte et réalise la coupe à mi-hauteur. Si les règles syntaxiques restent identiques, par contre les règles sémantiques subissent des modifications importantes, dues à la mise en place d'un attribut synthétisé "Mihaut". Il est associé aux non-terminaux M et D et a pour valeur la moyenne des luminosités des extrémités d'une monotonie qu'elle soit croissante (état M) ou décroissante (état D). La valeur de l'attribut "Mihaut" ne peut être connue qu'à la fin d'une monotonie, et comme elle intervient dans le calcul des attributs "Objet.P1" et "Objet.P3", on ne peut affecter une valeur à ces derniers que dans la mesure où "Mihaut" a sa valeur déterminée. Cette contrainte s'interprète comme un retour arrière sur le "ruban de lecture". L'automate de reconnaissance de la grammaire est donc un automate à pile (cf chap. V.2 Hiérarchie de Chomsky).

Règles syntaxiques	Règles sémantiques
ligne → pI   dI	Nlig(I) = ind. de ligne
ligne → mM	Nlig(M) = ind. de ligne Objet(M).Pl = j+1 Objet(M).Ll = luminosité(j) Objet(M).Lig = Nlig(M)
I → pI   dI	Nlig(I <sub>2</sub> ) = Nlig(I <sub>1</sub> )
I → #	-
I → mM	Nlig(M) = Nlig(I) Objet(M).Pl = j+1 Objet(M).Ll = luminosité(j) Objet(M).Lig = Nlig(M)
M → pM   mM	Mihaut(M <sub>1</sub> ) = Mihaut(M <sub>2</sub> ) Nlig(M <sub>2</sub> ) = Nlig(M <sub>1</sub> ) Objet(M <sub>2</sub> ).Ll = Objet(M <sub>1</sub> ).Ll Objet(M <sub>2</sub> ).Lig = Objet(M <sub>1</sub> ).Lig Objet(M <sub>2</sub> ).Pl = <u>si</u> luminosité(j) < Mihaut(M <sub>2</sub> ) <u>alors</u> j+1 <u>sinon</u> Objet(M <sub>1</sub> ).Pl <u>ssi</u>

M → #	-
M → dD	Mihaut(M) = (Objet(M).L1 + luminosité(j)) / 2 Nlig(D) = Nlig(M) Objet(D).P1 = Objet(M).P1 Objet(D).L1 = Objet(M).L1 Objet(D).Lig = Objet(M).Lig Objet(D).P2 = j Objet(D).L2 = luminosité(j) Objet(D).P3 = j
D → dD   pD	Mihaut(D <sub>1</sub> ) = Mihaut(D <sub>2</sub> ) Nlig(D <sub>2</sub> ) = Nlig(D <sub>1</sub> ) Objet(D <sub>2</sub> ).P1 = Objet(D <sub>1</sub> ).P1 Objet(D <sub>2</sub> ).L1 = Objet(D <sub>1</sub> ).L1 Objet(D <sub>2</sub> ).Lig = Objet(D <sub>1</sub> ).Lig Objet(D <sub>2</sub> ).P2 = Objet(D <sub>1</sub> ).P2 Objet(D <sub>2</sub> ).L2 = Objet(D <sub>1</sub> ).L2 Objet(D <sub>2</sub> ).P3 = <u>si</u> luminosité(j) >= Mihaut(D <sub>2</sub> ) <u>alors</u> j <u>sinon</u> Objet(D <sub>1</sub> ).P3 <u>fsi</u>
D → #	Objet(D).L3 = luminosité(j) Mihaut(D) = (Objet(D).L2 + luminosité(j)) / 2 <u>Si</u> ((CG(D) >= V) ou (CD(D) >= V)) <u>alors</u> Archivage d'Objet(D) <u>fsi</u>
D → mM	Nlig(M) = Nlig(D) Objet(D).L3 = luminosité(j) Mihaut(D) = (Objet(D).L2 + luminosité(j)) / 2 <u>si</u> ((CG(D) >= V) ou (CD(D) >= V)) <u>alors</u> Archivage d'Objet(D) <u>fsi</u> Objet(M).P1 = j+1 Objet(M).L1 = luminosité(j) Objet(M).Lig = Nlig(M)

Table 3

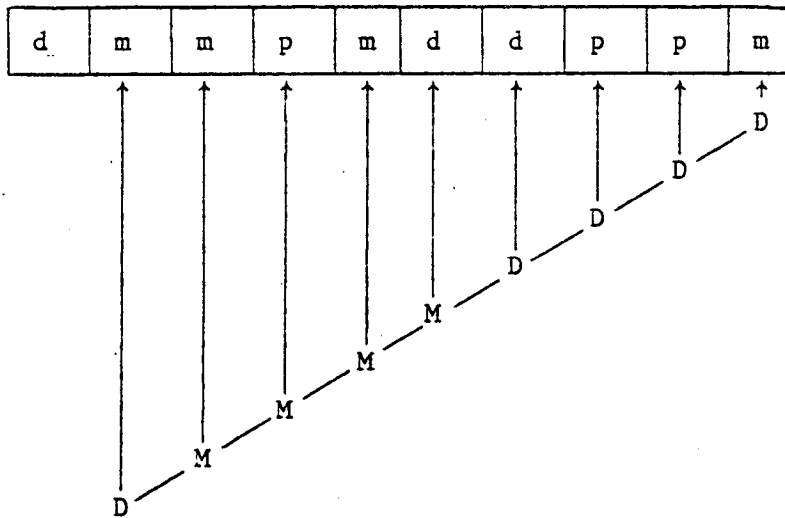
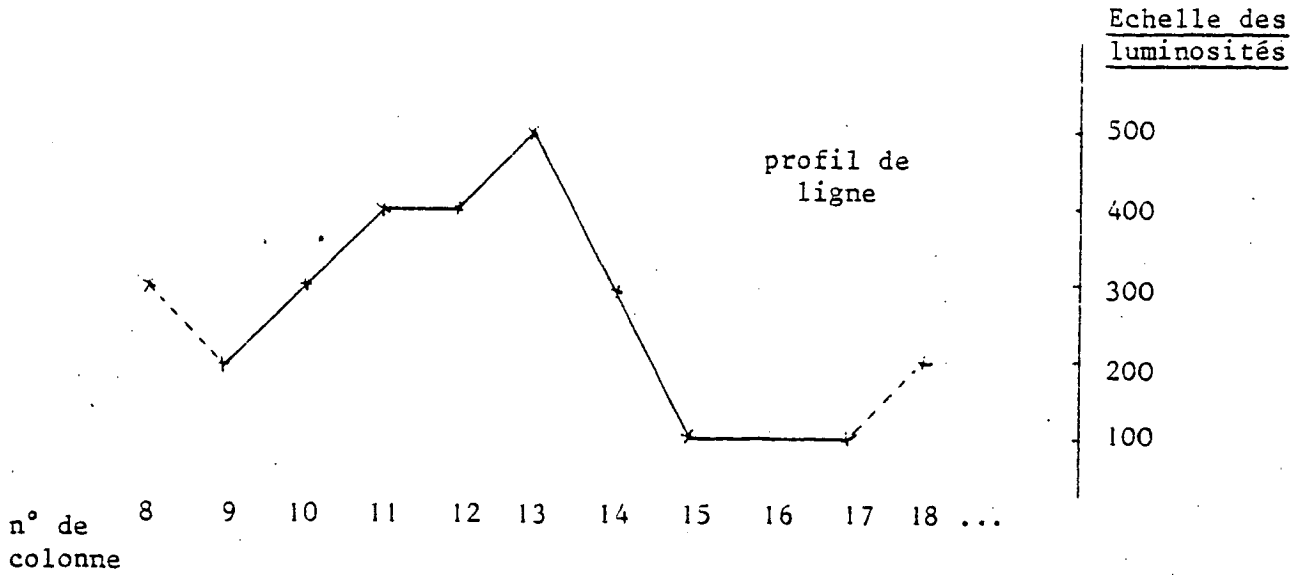
On peut remarquer la disparition de l'attribut correct de D. Ce dernier ne sert plus à rien car la coupe à mi-hauteur par définition positionne la fin d'un objet sur un point dont la luminosité est strictement supérieure à celle du point de la fin de la descente de l'objet.

La grammaire a été appliquée sur l'image non filtrée (fig. 13) au seuil de 5 ; si l'on descend ce seuil à 4, nous voyons que l'image devient trop bruitée, cela correspond à la critique 2 formulée au paragraphe "amélioration de la détection" (fig. 14).

Nous avons analysé l'image filtrée à l'aide de cette grammaire avec un seuil de 250, donc équivalent à un seuil de 4 de l'image initiale (fig. 15, fig. 16).

Tous les résultats présentés correspondent à la superposition à la visualisation des résultats lignes de l'image non transposée et de l'image transposée.

Exemple de fonctionnement avec coupe à mi-hauteur



x	x	x	x	x	x	x	x	x	Nlig
350	350	350	350	350	300	300	300	300	Mihaut

Valeur des attributs

x	x	x	x	x	x	x	x	x	Objet.Lig
200	200	200	200	200	200	200	200	200	Objet.L1
10	11	11	11	11	11	11	11	11	Objet.P1
				500	500	500	500	500	Objet.L2
				13	13	13	13	13	Objet.P2
								100	Objet.L3
				13	14	14	14	14	Objet.P3

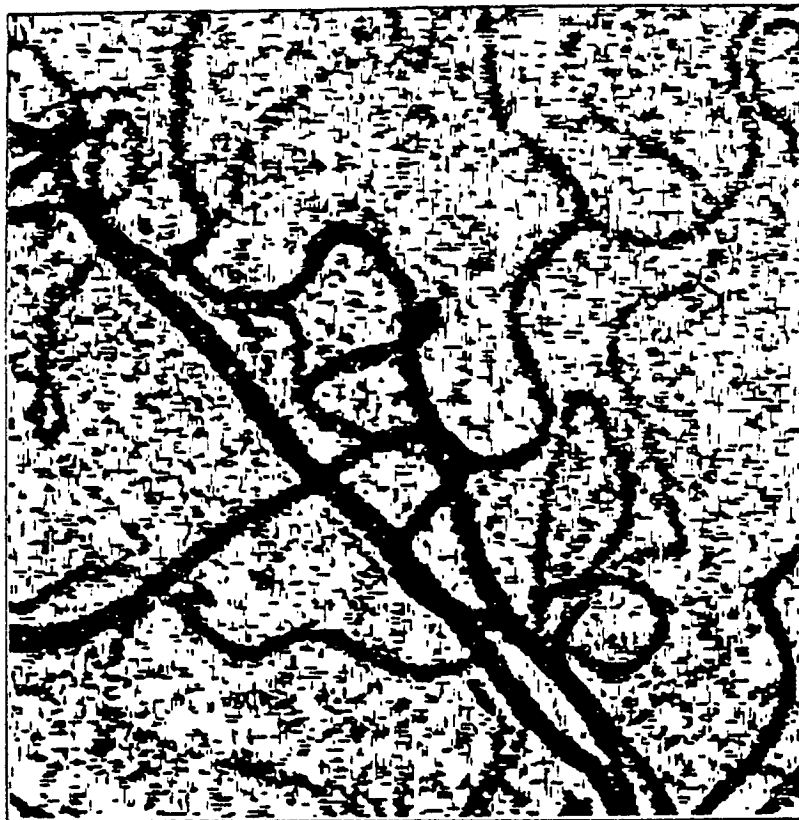


Figure 13 - Avant filtrage, seuil différentiel avec coupe à mi-hauteur : 5

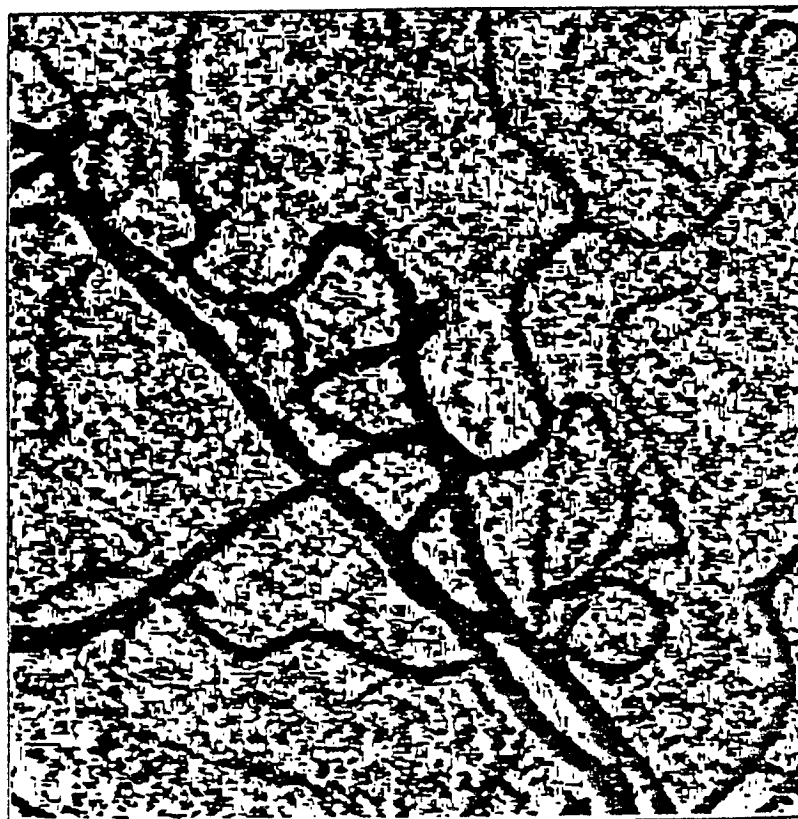


Figure 14 - Avant filtrage, seuil différentiel avec coupe à mi-hauteur : 4



Figure 15 - Après filtrage, seuil différentiel avec coupe à mi-hauteur : 250

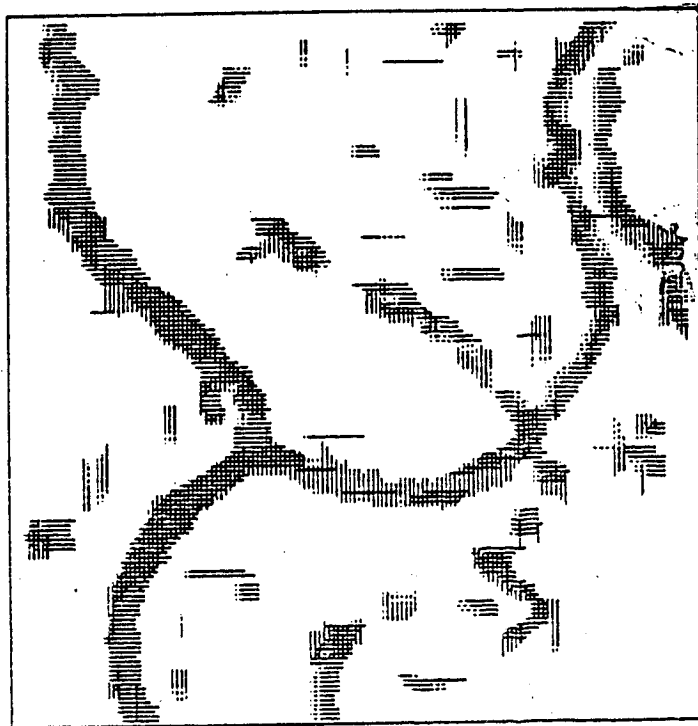


Figure 16 - Agrandissement du coin supérieur droit de la figure 15.

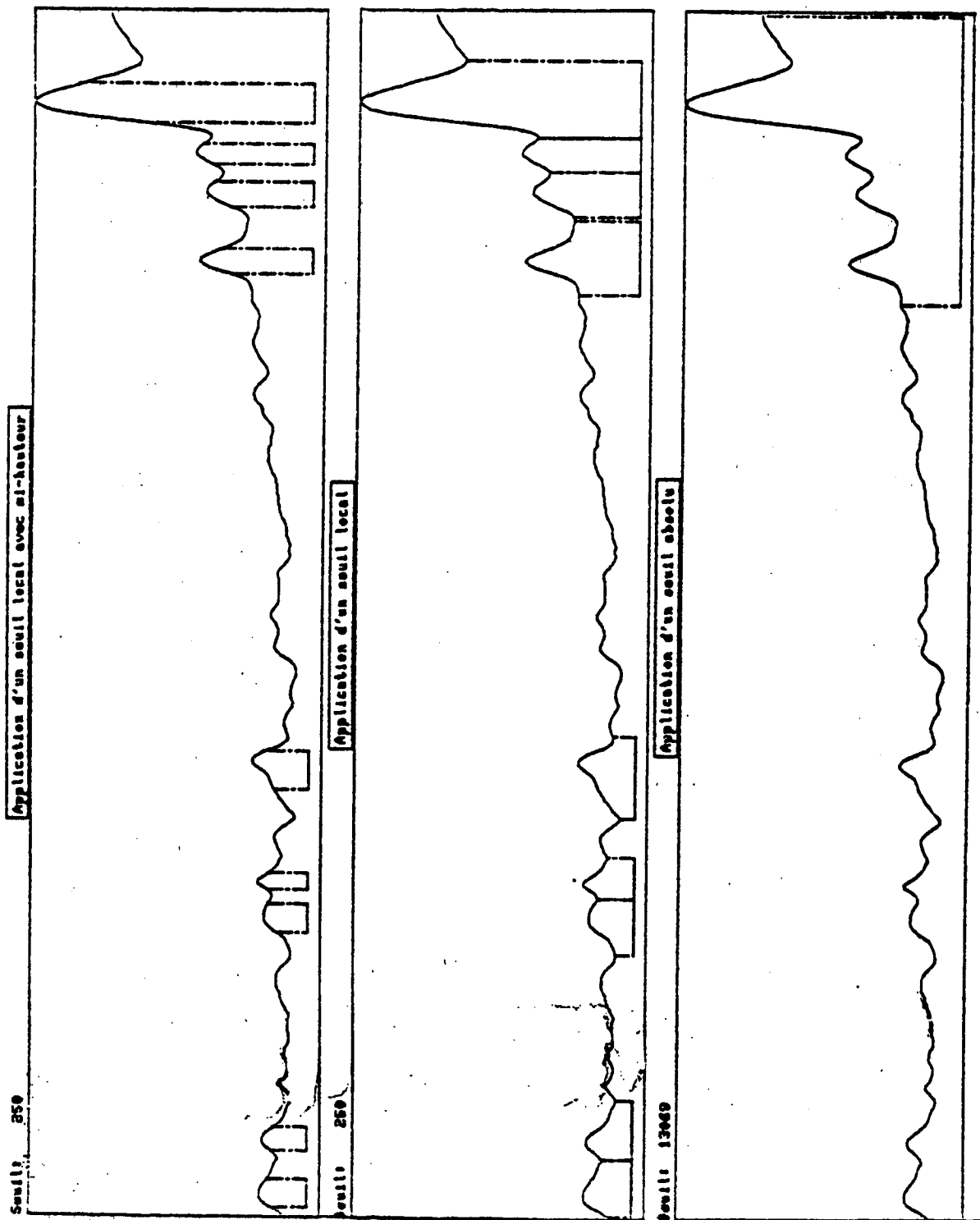


Figure 17 - Comparaison des 3 types de seuil.

# VIII - EXTENSION DE LA GRAMMAIRE A TOUTE L'IMAGE

Nous nous sommes définis des grammaires de segmentation d'une ligne d'image avec détection des objets à contraste suffisant.

En réalité, on peut définir une grammaire sur toute l'image et non pas seulement sur une de ses lignes.

Ajoutons quatre règles de production aux grammaires précédemment définies :

Image  $\rightarrow$  ligne | ligne S

S  $\rightarrow$  ligne | ligne S

Image représente l'image complète et devient l'axiome.

ligne représente une ligne d'image.

S représente une sous-image.

Ces quatre règles introduisent des règles sémantiques, précisant le fonctionnement de l'attribut hérité Nlig (Table 9)

Règles syntaxiques	Règles sémantiques
Image $\rightarrow$ ligne	Nlig (ligne) = 1
Image $\rightarrow$ ligne S	Nlig (ligne) = 1 Nlig (S) = 2
S $\rightarrow$ ligne	Nlig (ligne) = Nlig (S)
S <sub>1</sub> $\rightarrow$ ligne S <sub>2</sub>	Nlig (ligne) = Nlig (S <sub>1</sub> ) Nlig (S <sub>2</sub> ) = Nlig (S <sub>1</sub> ) + 1

Table 9

Les modifications à apporter aux règles sémantiques de la grammaire ligne, se limitent à transformer toutes les lignes de type "Nlig( ) = ind. de ligne" par "Nlig( ) = Nlig (ligne)".

Ce paragraphe montre donc que le formalisme grammaire d'attributs aurait pu être implanté sur toute l'image et non seulement sur une ligne.



## IX - DETECTEUR SYNTAXIQUE ISOTROPE A BALAYAGE LIGNE

Nous avons vu précédemment que pour opérer l'analyse verticale de l'image, nous transposions l'image et réappliquions l'analyse horizontale sur l'image transposée, la dernière étape consiste à fusionner les résultats dans un même référentiel dans le but d'une appréciation visuelle.

Nous allons développer dans ce paragraphe, une implémentation qui opère les analyses horizontales et verticales au cours d'un balayage ligne. Le traitement va donc devenir asymétrique au niveau implémentation.

### IX.1. Grammaire sans coupe à mi-hauteur

Les grammaires d'analyse ligne et colonne resteront rigoureusement identiques. Cependant, en réalité nous aurons (N+1) grammaires en fonctionnement, N étant le nombre de colonnes de l'image à analyser, soit une grammaire d'analyse ligne et une grammaire par colonne d'image. Ainsi l'analyse d'un point de coordonnées x et y, agira à la fois sur l'automate ligne et sur l'automate colonne de numéro x. Cette optique nous oblige à avoir (N+1) structures d'analyse, mais nous permet d'utiliser un balayage ligne de l'image et d'extraire en un seul passage les informations lignes et colonnes.

### IX.2. Grammaire avec coupe à mi-hauteur

La grammaire à coupe à mi-hauteur opère un retour-arrière pour trouver ses mi-hauteurs. Il est évident qu'un retour-arrière sur l'analyse ligne ne pose aucun problème, par contre sur l'analyse colonne, les mi-hauteurs sont recherchées sur les lignes précédentes. L'extraction des objets verticaux coupés à mi-hauteur nécessite donc de garder en mémoire un certain nombre de lignes pour pouvoir opérer ce retour arrière.

Ce nombre de lignes correspond au résultat du calcul (schéma 6)

$$\begin{matrix} \text{maximum} \\ \text{image} \end{matrix} (\text{dist}(\text{position mi-hauteur en montée}, \text{position sommet}) + 1, \\ \text{dist}(\text{position mi-hauteur en descente}, \text{position fin}) + 1)$$

Sur l'image, ce nombre de lignes nécessaires s'élève à une trentaine.

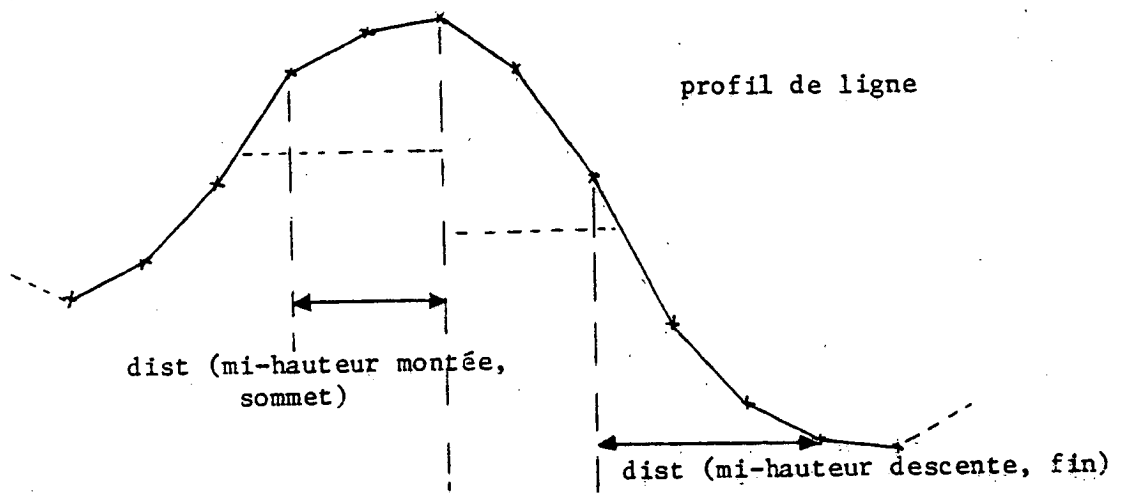


Schéma 6

### IX.3. Conclusion

En conclusion à cette partie, nous pouvons retenir que l'analyse horizontale et l'analyse verticale ont été effectuées par le même détecteur, mais avec des implémentations différentes.

## X - CONCLUSION

Les idées proposées donnent des résultats tout à fait satisfaisant sur les images traitées. Il est intéressant de remarquer que ces méthodes quoique syntaxiques sont a priori tout à fait concurrentielles par rapport aux méthodes numériques, non seulement au point de vue validité des résultats, mais aussi rapidité des calculs. Bien que nous n'ayons pas encore comparé notre méthode aux méthodes classiques, on peut évaluer sa vitesse par la table ci-dessous qui définit les besoins en calculs de l'algorithme sans coupe à mi-hauteur. De plus, la programmation du détecteur isotrope peut se faire en suivant un balayage ligne avec en mémoire centrale, deux lignes d'image si l'on n'effectue pas la coupe à mi-hauteur, avec une trentaine de lignes sinon.

	Analyse ligne	Analyse colonne
Pour chaque pixel	1 test	2 tests
Pour chaque forme caractéristique	2 soustractions 2 tests	2 soustractions 2 tests

Table 10

Remerciements : Ce travail a été rendu possible grâce au soutien apporté par le Professeur SCARABIN du CHR de Rennes et aux contacts que nous avons développés avec son équipe, tout particulièrement Mr GIBAUD.

## BIBLIOGRAPHIE

- [Bout] BOUTHEMY  
Modélisation et suivi de perturbations nuageuses en imagerie météorologique.  
Thèse de docteur-ingénieur 1982 U.E.R. Mathématiques et informatique  
Rennes I.
- [Cam1] CAMILLERAPP  
L'utilisation du contexte en reconnaissance des formes.  
Thèse d'état 1976 Paris VI Mathématiques et informatique.
- [Cam2] CAMILLERAPP - CUZON  
Utilisation du contexte pour la détection d'objets faiblement contrastés.  
Congrès AFCET - 1979 Tome II, p. 217-224.
- [Cam3] CAMILLERAPP - LEPLUMEY  
Codage du contexte et détection au moyen de règles de réécriture.  
Congrès AFCET - 1984 Tome I, p. 93-100.
- [Cuz] CUZON  
Reconnaissance des vaisseaux sur une radiographie crânienne.  
Rapport de DEA, INSA Rennes, juin 1978.
- [Dud] DUDA & HART  
Pattern classification and scene analysis.  
Ed. Wiley - Interscience Publication 1973, p. 356-362, p. 425-435.
- [Ful] FU  
Digital Pattern Recognition.  
Syntactic (linguistic) Pattern recognition.  
Springer-Verlag, Berlin Heidelberg, 1976, p. 95-134.
- [Fu2] FU  
Syntactic methods in pattern recognition.  
Mathematics in science and engineering, Volume 12, Academic Press 1974.

- [Gas] GASNIER  
Détection et suivi de contours : Etude comparative de méthodes séquentielles et application à des radiographies et des images TV.  
Thèse de 3ème cycle, 1980, UER Mathématiques et Informatique  
Rennes I.
- [Gon] GONZALEZ - THOMASON  
Syntactic Pattern Recognition, an introduction.  
Addison-Wesley Publishing Company, 1978.
- [Hor] HOROWITZ  
Peak recognition in Waveforms.  
Syntactic Pattern Recognition : Applications,  
Ed. by K.S. FU, Springer Verlag, 1977.
- [Kac1] KACED  
Le point sur les k-formes. Définition et applications nouvelles  
en traitement d'images.  
Congrès AFCET 1984, Tome I, p. 93-100.
- [Kac2] KACED - MOHR - SCHNELL - HATON  
Application de méthodes de traitements numériques d'images à  
l'examen des chaussées routières.  
Congrès AFCET 1979, Tome II, p. 174-184.
- [Knu] KNUTH  
Semantic of context-free languages.  
J. Math. Syst. Theory, vol. 2, pp. 127-146, 1968.
- [Liv] LIVERCY  
Théorie des programmes.  
Dunod Informatique.
- [Mait1] MAITRE - FAUST  
Détection des contours d'image dans un but de reconnaissance des  
formes.  
Congrès AFCET 1978, Tome I, p. 456-465.

- [Mait2] MAITRE - ATTAL - BENIGOT  
Une méthode arborescente de lecture automatique de gels de séquences de nucléotides.  
Congrès AFCET 1984, Tome I, p. 531-538.
- [Mic] MICLET  
Méthodes structurelles pour la reconnaissance des formes.  
Eyrolles, collection technique et scientifique des télécommunications (CNET ENST), juin 1984.
- [Pag] PAGAN  
Formal specification of programming languages : a panoramic primer.  
Prentice Hall, INC Englewood Cliffs New Jersey 1981.
- [Pel] PELI - MALAH  
A study of edge detection algorithms .  
Computer graphics and image processing, volume 20, 1982, p. 1-21.
- [Quig] QUIGNON - FAURE  
Utilisation de méthodes descriptives pour l'interprétation automatique de l'électromyogramme.  
Congrès AFCET 1984, Tome I, p. 423-431.
- [Quinq] QUINQUETON  
Recherche d'alignements dans une image de points.  
Thèse de 3e cycle, juin 1976, Paris VI, Mathématiques et informatique.
- [Ram] RAMESH JAIN - DOUG RHEAUME  
A two stage method for fast edge detection.  
Computer graphics and image processing, Volume 14, 1980, p. 177-181.
- [Set] SETHI  
Edge detection using charge analogy.  
Computer graphics and image processing, volume 20, 1982, p. 185-195.

[Sim]

SIMON

La reconnaissance des formes par algorithmes.

Masson, Etudes et recherche en informatique, 1984.

[Tim]

TIMOUNI

Détection de contours.

Cours et séminaires : vision par ordinateur, INRIA Rocquencourt,  
juin 1983.

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique

9.

7

9.

2.

3)

0.